



# Estimation récursive de primitives 3D au moyen d'une caméra mobile

Patrick Rives, Bernard Espiau

## ► To cite this version:

Patrick Rives, Bernard Espiau. Estimation récursive de primitives 3D au moyen d'une caméra mobile. [Rapport de recherche] RR-0652, INRIA. 1987. inria-00075901

**HAL Id: inria-00075901**

**<https://inria.hal.science/inria-00075901>**

Submitted on 24 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNITÉ DE RECHERCHE  
INRIA-RENNES

Institut National  
de Recherche  
en Informatique  
et en Automatique

Domaine de Voluceau  
Rocquencourt  
BP 105  
78153 Le Chesnay Cedex  
France

Tél (1) 39 63 55 11

Rapports de Recherche

N° 652

**ESTIMATION RÉCURSIVE  
DE PRIMITIVES 3 D  
AU MOYEN  
D'UNE CAMÉRA MOBILE**

Patrick RIVES  
Bernard ESPIAU

Mars 1987

Campus Universitaire de Beaulieu  
35042 - RENNES CÉDEX  
FRANCE  
Téléphone: 99 36 20 00  
Télex: UNIRISA 950 473 F  
Télécopie: 99 38 38 32

Publication Interne n° 350

Mars 1987 - 36 pages

### RECURSIVE ESTIMATION OF 3D FEATURES FOR MOBILE VISION SYSTEM ESTIMATION RECURSIVE DE PRIMITIVES 3D AU MOYEN D'UNE CAMERA MOBILE

Patrick Rives, Bernard Espiau  
IRISA - Centre INRIA de Rennes  
Campus de Beaulieu  
35042 Rennes Cedex

**Résumé :** Ce rapport présente un schéma d'estimation de primitives géométriques 3D à partir d'informations contenues dans une séquence d'images fournie par une caméra mobile dont le mouvement est connu. Nous rappelons tout d'abord les modèles de bases utilisés: points et segments de droites et nous établissons comment leur projection perspective dans l'image varie avec le mouvement de la caméra. Des techniques de filtrage récursif au sein de la séquence d'images sont utilisées pour reconstruire, de façon incrémentale, la scène 3D, au fur et à mesure du déplacement de la caméra. Des résultats expérimentaux dans le domaine de la robotique sont présentés.

**Mots clés :** *mouvement, séquence d'images, Optic Flow, filtrage récursif, boucle fermée, robotique.*

**Abstract :** This technical report presents a scheme allowing to estimate parameters which describe geometrical structures in a 3D scene by only using informations issued from a sequence of images provided by a mobile vision sensor with known motion. We first recall the basic used models: points and lines, and then we relate their perspective projection in the image plane to the camera motion. Some techniques of recursive filtering are used into the sequence of images to incrementally build the 3D scene all along the displacement of the camera. Some experimental results in the field of robotics are given.

**Key words :** *motion, time varying analysis, Optic Flow, recursive filtering, closed loop control, robotics.*

# 1 Introduction

Un récent développement en vision par ordinateur et analyse de scène concerne l'utilisation explicite du *mouvement* pour estimer la structure géométrique et cinématique des primitives 3D contenues dans la scène. Ce mouvement peut être introduit dans les algorithmes tant dans le domaine de l'image (champ des vitesses apparentes dans l'image, connu sous le nom d'*Optic Flow*, résultant de la projection du champ des vitesses dans la scène), que dans le domaine de la scène (mouvements dus à la caméra et/ou aux objets mobiles dans la scène).

L'estimation du mouvement dans l'image ou, plus exactement dans la *séquence* d'images, peut être utilisée dans des buts de segmentation ou pour accéder à des informations 3D au niveau de la scène dans ce dernier cas des contraintes supplémentaires sur celle-ci doivent être imposées. Pour estimer le champ des vitesses dans l'image, deux classes d'approches ont été mises en oeuvre : des techniques de mises en correspondances d'éléments caractéristiques dans la suite d'images (*features based methods*) comme dans [1], [2], [3] ou [4] et des techniques différentielles (*gradient based methods*) comme décrites dans [5], [6], [7], [8], [9]. Pour exploiter cette estimation du mouvement dans l'identification des structures géométriques dans la scène, il est nécessaire d'établir un modèle des primitives utilisées (points, courbes, surfaces ...) non seulement géométrique mais aussi cinématique et d'étudier les propriétés de la projection de ce modèle dans la séquence d'images [10], [11], [12].

L'analyse du mouvement peut se révéler très utile dans de nombreuses applications, principalement en robotique : stéréovision [13], [4], identification de trajectoire dans le cas d'un robot mobile ou d'une cible [10], [14], contrôle adaptatif de robots [15] et certaines tâches spécifiques de robotique comme, par exemple, la navigation et l'évitement d'obstacles [16].

Ce rapport présente un schéma général d'estimation de primitives 3D basé sur l'utilisation explicite du mouvement à la fois dans la séquence d'images et dans la scène. Ce schéma est appliqué à l'estimation de la profondeur relative caméra-objets (*carte des profondeurs*) puis étendu à des primitives 3D de type segments. Nous restreindrons l'étude au cas où le capteur est en mouvement dans un univers statique.

L'idée de base de ce schéma est d'utiliser des mesures instantanées de paramètres 2D dans la séquence d'images pour raffiner l'estimation sur les primitives 3D au fur et à mesure du déplacement du capteur dans la scène. De plus, comme nous contrôlons ce déplacement, cela nous permettra de l'optimiser en fonction de la trajectoire nominale désirée dans la tâche principale (par exemple, trajectoire de référence) et du mouvement entraînant une meilleure estimation de la primitive 3D. C'est une approche typique de commande en *boucle fermée*.

Le rapport est organisé comme suit : la Section II est consacrée à l'établissement des modèles cinématiques reliant l'aspect géométrique des primitives 3D aux mouvements de la caméra. La Section III est dédiée à l'extraction des paramètres dynamiques dans la séquence d'images. Dans la Section IV, nous décrivons l'aspect estimation récursive de la primitive 3D tout au long de la séquence d'images. La Section V est consacrée à l'élaboration de la commande en

boucle fermée du robot vis à vis de son environnement. Enfin, la Section VI présentera les résultats obtenus en simulation dans le cadre d'une tâche d'évitement d'obstacles.

## 2 Modélisation des primitives :

### 2.1 Hypothèses de bases :

Nous considérons dans un espace 3D, un ensemble d'objets statiques. Un repère de référence  $R_s$  est associé à la scène 3D, un repère  $R_c$  est lié à la caméra mobile dont le modèle est du type projection centrale et la distance focale choisie égale à un (Fig. 1).

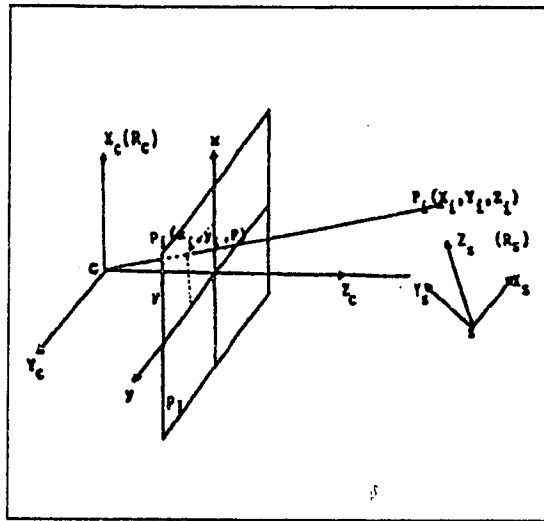


Figure 1. Modèle de la caméra

Les coordonnées du point  $P$  dans  $R_s$  sont :

$$X_s^T = (X_s \ Y_s \ Z_s) \quad (1)$$

et, dans  $R_c$  :  $X^T = (X, Y, Z), \quad Z > 1 \quad (1')$

Sa projection dans le plan image est, en utilisant les coordonnées homogènes :

$$\begin{pmatrix} x' \\ y' \\ z' \\ \omega \end{pmatrix} = \begin{pmatrix} I_3 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2)$$

avec 
$$\underline{x}^P = \frac{1}{\omega} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} \quad (3)$$

nous obtenons donc les coordonnées images (avec  $F=1$ )

$$\begin{cases} x = \frac{X}{Z} \end{cases} \quad (4)$$

$$\begin{cases} y = \frac{Y}{Z} \end{cases} \quad (5)$$

Le mouvement de la caméra est complètement décrit par le torseur cinématique associé  $\underline{\tau}$ , qui sera exprimé dans le repère  $R_c$  et noté :

$$\underline{\tau} = \begin{pmatrix} -\underline{V} \\ -\underline{\Omega} \end{pmatrix} \quad (6)$$

avec  $\underline{V}^T = (V_X \ V_Y \ V_Z)$  (7)

$$\underline{\Omega}^T = (\Omega_X \ \Omega_Y \ \Omega_Z)$$
 (8)

ce qui permet d'exprimer la vitesse relative de n'importe quel point  $P$  dans  $R_c$  par :

$$\frac{d\underline{X}}{dt} = \dot{\underline{X}} = \underline{V} + \underline{\Omega} \times \underline{X} \quad (9)$$

où  $\times$  représente le produit vectoriel.

## 2.2 Relations entre les modèles 3D et les paramètres dynamiques dans la séquence d'images

L'estimation de la primitive 3D est réalisée à partir de mesures dans la séquence d'images, il est donc nécessaire d'établir les équations reliant les paramètres de la primitives à des paramètres (2D + t) extraits de la séquence d'images.

### 2.2.1 Cas d'une primitive de type "point"

Les paramètres de la primitive 3D de type "point", sont les coordonnées 3D du point dans le repère  $R_c$ . Les équations reliant ces coordonnées à des mesures dans le repère image sont obtenues en dérivant (eqs 4 et 5) et en utilisant (eq. 9), soit :

$$v = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} ; \quad \begin{cases} \dot{x} = u_T + u_R \\ \dot{y} = v_T + v_R \end{cases} \quad (10)$$

$$(11)$$

$$\text{avec} \quad \begin{cases} u_T = \frac{1}{Z} (-V_X + xV_Z) \end{cases} \quad (12)$$

$$\begin{cases} v_T = \frac{1}{Z} (-V_Y + yV_Z) \end{cases} \quad (13)$$

$$\begin{cases} u_R = y\Omega_Z + x\Omega_X - (1+x^2)\Omega_Y \end{cases} \quad (14)$$

$$\begin{cases} v_R = -x\Omega_Z - x\Omega_Y + (1+y^2)\Omega_X \end{cases} \quad (15)$$

A partir de ces expressions, il apparait que la connaissance de  $\tau$  peut permettre de calculer les coordonnées 3D (X,Y,Z) d'un point P dans le repère  $R_c$ , si on peut mesurer dans l'image une forme linéaire :

$$v^* = \lambda_1 \dot{x} + \lambda_2 \dot{y} \quad (16)$$

où  $\lambda_1$  et  $\lambda_2$  sont connus.

Une condition supplémentaire, due aux (eqs 12 et 13) est naturellement :

$$\|v\| \neq 0 \quad (17)$$

Dans la section suivante, nous montrerons comment estimer  $v^*$  qui représente le vecteur des vitesses apparentes dans l'image (*Optic Flow*).

### 2.2.2 Cas d'une primitive de type "segment"

Considérons un segment de droite  $\mathcal{D}$  dans  $R_c$ , défini par son équation paramétrique :

$$\underline{X} = \underline{X}_0 + \xi \underline{u} \quad , \quad \forall \xi \in \mathcal{D} \quad (18)$$

ou, de façon équivalente, par :

$$(\underline{X} - \underline{X}_0) \times \underline{u} = 0 \quad (19)$$

La représentation de  $\mathcal{D}$  par  $\underline{X}_0$  et le vecteur directeur  $\underline{u}$  est redondante. Pour établir le modèle cinématique d'un segment en mouvement, nous choisissons de le représenter par le couple  $(\underline{u}, \underline{h})$ , où :

$$\underline{h} = \underline{X}_0 \times \underline{u} \quad (20)$$

avec

$$\underline{u}^T = (\alpha \ \beta \ \gamma) \quad (21)$$

Les composantes de  $\underline{h}(h_1, h_2, h_3)$  sont également liées par l'équation :

$$\langle \underline{h} \cdot \underline{u} \rangle = 0 \quad (22)$$

Il est alors possible d'établir le modèle cinématique dans  $R_c$  en différentiant (eq. 19) par rapport au temps ce qui conduit à la relation :

$$\dot{\underline{X}} \times \underline{u} - \dot{\underline{X}}_0 \times \underline{u} - \underline{u} \times (\underline{X} - \underline{X}_0) = 0 \quad (23)$$

en utilisant (eqs 9 et 18), nous obtenons :

$$(\underline{V} - \dot{\underline{X}}_0 + \underline{\Omega} \times \underline{X}_0) \times \underline{u} + \xi(\underline{\Omega} \times \underline{u} - \dot{\underline{u}}) \times \underline{u} = 0 \quad (24)$$

Cette relation est vraie pour tout  $\zeta$  et il est aisé de montrer qu'une condition nécessaire et suffisante est :

$$\left\{ \begin{array}{l} (\dot{\underline{u}} - \underline{\Omega} \times \underline{u}) \times \underline{u} = 0 \\ (\underline{V} + \underline{\Omega} \times \underline{X}_0 - \dot{\underline{X}}_0) \times \underline{u} = 0 \end{array} \right. \quad (25)$$

$$\quad \quad \quad (26)$$

dont on peut extraire deux équations indépendantes pour  $\underline{u}$ .

Comme  $\underline{u}$  est défini avec un degré de liberté, nous pouvons rajouter une contrainte (par exemple  $\|\underline{u}\| = 1$ ) qui permet de définir complètement l'équation (eq. 25) et de l'écrire sous la forme :

$$\dot{\underline{u}} = \mu(\underline{u}, \underline{\Omega}) \underline{u} + \underline{\Omega} \times \underline{u} \quad (27)$$

où  $\mu(u, \Omega)$  est un scalaire déterminé par le choix de la contrainte.



L'équation (eq. 26) peut être réécrite :

$$\underline{V} \times \underline{u} - \dot{\underline{X}}_0 \times \underline{u} + (\underline{\Omega} \times \underline{X}_0) \times \underline{u} = 0 \quad (28)$$

la dérivation de (eq. 20) donne :

$$\dot{\underline{h}} = \dot{\underline{X}}_0 \times \underline{u} + \underline{X}_0 \times \dot{\underline{u}} \quad (29)$$

qui, une fois substituée dans (eq. 28) et en utilisant (eq. 27), amène à :

$$\dot{\underline{h}} = \underline{V} \times \underline{u} + \mu \underline{h} + \underline{X}_0 \times (\underline{\Omega} \times \underline{u}) + (\underline{\Omega} \times \underline{X}_0) \times \underline{u} \quad (30)$$

soit, enfin :

$$\dot{\underline{h}} = \underline{V} \times \underline{u} + \mu \underline{h} - \underline{h} \times \underline{\Omega} \quad (31)$$

Cette équation nous permet d'extraire deux équations indépendantes en utilisant la contrainte fournie par l'équation (eq. 22). Nous avons donc établi quatre équations différentielles indépendantes (eqs 27 et 31) décrivant complètement la variation des paramètres représentant la droite  $\mathcal{D}$  due au mouvement relatif de  $R_c$  par rapport à  $R_s$ .

Les calculs précédents ont permis d'établir un modèle cinématique de primitive 3D de type "segment", il est nécessaire de relier les paramètres de ce modèle à des mesures dans la séquence d'images.

La projection de  $\mathcal{D}$  dans le repère image est un segment de droite  $D$  obtenu en éliminant  $Z$  dans deux des équations fournies par (eq. 19). Si on considère le cas où  $\gamma \neq 0$ , l'équation de  $D$  est de la forme :

$$Px + 1 = qy \quad (32)$$

où  $p$  et  $q$  sont donnés par :

$$\begin{cases} Ph_2 + qh_1 = 0 \\ \beta q - \alpha p = \gamma \end{cases} \quad (33)$$

$$(34)$$

Nous allons démontrer maintenant que si nous pouvons estimer  $(p, q, \dot{p}, \dot{q})$ , alors il est possible de calculer les paramètres  $(\underline{u}, \underline{h})$  qui caractérisent  $\mathcal{D}$ .

Pour cela, nous choisissons de fixer la contrainte sur  $\underline{u}$  telle que :

$$\|\underline{u}\| = 1 \quad (35)$$

Ce choix entraîne que  $\dot{\underline{u}}$  est orthogonal à  $\underline{u}$ , donc :

$$\underline{u}^T (\underline{\Omega}, \underline{u}) = 0 \quad (36)$$

l'équation (eq. 27) devient alors :

$$\dot{\underline{u}} = \underline{\Omega} \underline{X} \underline{u} \quad (37)$$

Pour faciliter l'écriture, posons :

$$\underline{s} = \{-p \quad q \quad -1\}^T \quad (38)$$

ce qui permet d'écrire l'équation (eq. 34) sous la forme :

$$\underline{s}^T \underline{u} = 0 \quad (39)$$

En différentiant (eq. 39) et en utilisant (eq. 37), on obtient :

$$\underline{s}^T (\underline{\Omega} \underline{X} \underline{u}) + \dot{\underline{s}}^T \underline{u} = 0 \quad (40)$$

qui peut être écrite (en appelant  $\omega$  la matrice antisymétrique associée au produit vectoriel  $\underline{X} \underline{\Omega}$ ) :

$$\begin{pmatrix} \underline{s}^T \omega - \dot{\underline{s}}^T \\ \underline{s}^T \end{pmatrix} \underline{u} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ -p & q & -1 \end{pmatrix} \underline{u} = 0 \quad (41)$$

avec

$$\begin{cases} a_{11} = -q\Omega_Z - \Omega_Y + \dot{p} \\ a_{12} = -p\Omega_Z + \Omega_X - \dot{q} \\ a_{13} = p\Omega_Y + q\Omega_X \end{cases} \quad (42)$$

$$\quad \quad \quad (43)$$

$$\quad \quad \quad (44)$$

En posant

$$A = qa_{11} + pa_{12} \quad (45)$$

et

$$F = qa_{13} - p \quad (46)$$

l'équation (eq. 41) est alors équivalente à

$$\begin{pmatrix} A & 0 & F \\ -P & q & -1 \end{pmatrix} \underline{u} = 0 \quad (47)$$

(48)

qui donne

$$\alpha = \frac{F}{A} \gamma \quad (49)$$

et

$$\beta = \frac{PF + A}{qA} \gamma \quad (50)$$

Finalement, en utilisant (eq. 35), nous obtenons l'expression des deux premières composantes de  $\underline{u}$

$$\alpha = \frac{qAF}{\sqrt{q^2 F^2 (A^2 + F^2) + A^2 (PF + A)^2}} \quad (51)$$

et

$$\beta = \frac{PF + A}{qF} \alpha \quad (52)$$

Nous allons appliquer une démarche identique pour exprimer les composantes de  $\underline{h}$  en fonction de  $(p, q, \dot{p}, \dot{q})$ .

Posons

$$\underline{\tau} = (q \quad P \quad 0)^T \quad (53)$$

alors, l'équation (eq. 33) devient :

$$\underline{\tau}^T \underline{h} = 0 \quad (54)$$

En différentiant (eq. 54) et en utilisant (eqs. 31 et 36), on obtient :

$$(\dot{\underline{\tau}}^T - \underline{\tau}^T \omega) \underline{h} + \underline{\tau}^T \underline{V} \underline{x}_u = 0 \quad (55)$$

En utilisant la propriété

$$\underline{\tau}^T \underline{V} \underline{X} \underline{u} = \underline{\gamma} \underline{s}^T \underline{V} \quad (56)$$

alors, les équations (eqs. 22, 54, 55) forment le système suivant :

$$\{b_{ij}\} \underline{h} = \begin{pmatrix} \tau^T \omega - \dot{\tau}^T \\ \tau^T \\ u^T \end{pmatrix} \underline{h} = \begin{pmatrix} \underline{\gamma} \underline{s}^T \underline{V} \\ 0 \\ 0 \end{pmatrix} \quad (57)$$

avec  $s^T V \neq 0$  pour une solution non nulle (condition analogue à celle donnée par (eq. 17)).

En remarquant que

$$\begin{cases} b_{11} = a_{12} - \Omega_X \\ b_{12} = -a_{11} - \Omega_Y \\ b_{13} = -q\Omega_Y + P\Omega_X \end{cases} \quad (58)$$

$$\quad \quad \quad (59)$$

$$\quad \quad \quad (60)$$

et en utilisant (eq. 45), le système s'exprimera :

$$\begin{pmatrix} 0 & A - b_{13} & -qb_{13} \\ q & P & 0 \\ \alpha & \beta & \gamma \end{pmatrix} \underline{h} = \begin{pmatrix} -q\gamma s^T V \\ 0 \\ 0 \end{pmatrix} \quad (61)$$

dont la solution est

$$h_2 = \frac{q\alpha}{P} \underline{s}^T \underline{V} \quad (62)$$

avec  $h_1$  obtenu par (eq. 54) et  $h_2$  par (eq. 22).

**Remarque**

Dans le cas particulier d'une translation du capteur, le système complet définissant  $(\underline{u}, \underline{h})$  sera :

$$\left\{ \begin{aligned} \frac{\alpha}{\beta} &= \frac{\dot{q}}{P} \end{aligned} \right. \quad (63)$$

$$\left\{ \begin{aligned} \alpha &= \frac{P - q \frac{\dot{P}}{\dot{q}}}{\sqrt{1 + \left(1 + \frac{P}{q}\right)^2} \left(P - \frac{\dot{P}}{q}\right)^2} \end{aligned} \right. \quad (64)$$

$$\left\{ \begin{aligned} h_2 &= \frac{q}{P} \alpha (-PV_X + qV_Y - V_Z) \end{aligned} \right. \quad (65)$$

$$\left\{ \begin{aligned} h_1 &= -\frac{P}{q} h_2 \end{aligned} \right. \quad (66)$$

### 3 Extraction des mesures dans la séquence d'images

Nous venons d'établir les équations reliant le mouvement dans le plan image aux paramètres caractérisant des primitives 3D de type "points" et "segments", cette section est consacrée à l'extraction de ce mouvement dans la séquence d'images. Dans le cas de primitives "points", nous utiliserons une approche de type *Optic Flow* et dans le cas des segments, une approche de type *approximation polygonale*

#### 3.1 Cas des primitives "points"

##### 3.1.1 Estimation du mouvement dans l'image (Optic Flow)

Le champ des vitesses apparentes dans l'image (*Optic Flow*) est défini comme le champ de vecteurs mettant en correspondance les pixels entre deux images prises à des instants différents et après déplacement du capteur et/ou des objets dans la scène. Dans la pratique, en imagerie numérique, on assimilera la mesure du champ de vitesse à une mesure de déplacement.

Une hypothèse couramment utilisée consiste à poser que la variation de l'intensité dans la scène est uniquement due au mouvement de la caméra ce qui permet d'écrire l'équation bien connue liant le gradient spatial et les composantes de l'Optic Flow dans l'image.

$$\frac{dI}{dt} = 0 \quad (67)$$

$$\langle \underline{\nabla} I \cdot \underline{v} \rangle + \frac{\partial I}{\partial t} = 0 \quad (68)$$

ou'  $\langle . \rangle$  représente le produit scalaire.

Cette équation montre que, localement, seule la composante  $\underline{v}^\perp$  orthogonale au contour dans l'image est mesurable. Cependant, si nous sommes capables d'estimer à la fois  $\underline{v}^\perp$  et la direction du gradient spatial, alors nous pourrions déterminer  $(\lambda_1, \lambda_2)$  dans (eq. 16) et le calcul de la carte des profondeurs deviendra possible.

Parmi les nombreuses méthodes citées en référence, permettant d'extraire la composante orthogonale de l'optic flow, l'algorithme décrit dans [8] est un des plus efficace. A partir d'une modélisation spatio-temporelle d'un élément de contour en mouvement, une technique de maximum de vraisemblance discret permet de déterminer un ensemble de masques qui constitue un EOF (Estimateur d'Optic Flow). Pour chaque pixel appartenant à un contour, cet opérateur fournit l'estimation de (Fig 2) :

- l'orientation spatiale du contour,  $\tilde{\theta}$
- la composante orthogonale au contour de l'optic flow  $\tilde{\underline{v}}^\perp$ , avec la valeur de vraisemblance associée.

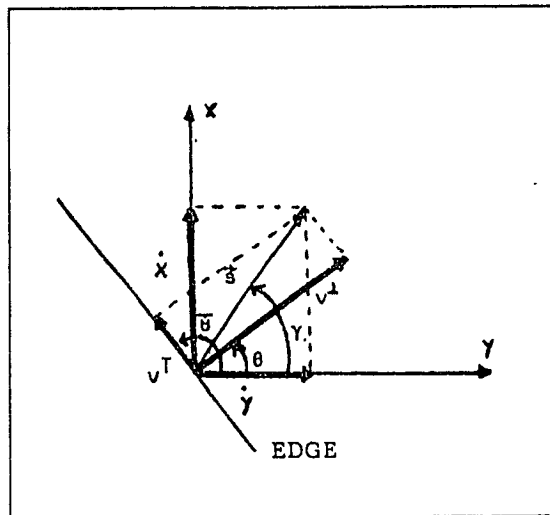


Figure 2. Estimation locale de l'optic flow

L'équation (eq. 16) devient alors :

$$\tilde{\underline{v}}^\perp = \tilde{\lambda}_1 \dot{\underline{y}} + \tilde{\lambda}_2 \dot{\underline{x}} \quad (69)$$

avec

$$\tilde{\lambda}_1 = \cos \tilde{\theta} \quad (70)$$

$$\tilde{\lambda}_2 = \sin \tilde{\theta} \quad (71)$$

Nous utiliserons cet opérateur dans la section suivante.

## 3.2 Cas des primitives "segments"

### 3.2.1 Estimation des segments dans l'image

L'estimation de segments dans l'image peut être décomposée en deux phases: la détection des contours spatiaux et leur approximation polygonale.

Concernant l'extraction de contours, de nombreuses méthodes peuvent être utilisées comme les techniques de gradient ou de laplacien [17], [18], ou de filtrage récursif [19], [20]; dans de nombreux cas, des étapes d'amincissement et de chainage seront nécessaires pour obtenir des contours de longueur maximum.

La deuxième étape consiste à approximer le contour par des segments de droites. Le principe de base consiste à chercher la meilleure approximation d'un nuage de points par un segment de droite ce qui est équivalent à minimiser :

$$\hat{d}^2 = \min_{P,q} \{d^2 = \sum_{i=1}^n d_i^2\} \quad (72)$$

avec  $n$  égal au nombre de pixel et  $d_i$  la "distance" du pixel  $i$  à la droite cherchée dans le cas d'une distance Euclidienne :

$$d_i^2 = \frac{(Px_i - y_i + q)^2}{1 + p^2} \quad (73)$$

La précision de l'approximation est alors donnée par la valeur du résidu.

Cette approche peut être directement étendu à l'approximation d'une courbe en utilisant l'algorithme itératif suivant :

*Begin*

$k = 1$

*For*  $i = 3, n$  le long de la courbe ( $n$  nombre de points de la chaîne)

calculer  $(p,q)$  tel que  $\hat{d}^2 = \min_{P,q} \{d^2 = \sum_{j=k}^{i-1} d_j^2\}$

calculer  $d_i^2$

Si  $d_1^2 > d_{\text{threshold}}$

Alors

*i est un sommet du polygone*

*réinitialiser  $k = i$*

Sinon

Fin

Fin

Le principal avantage de ce type d'algorithme comparé à des méthodes globales ("divide and conquer"[21]), résulte dans sa possibilité d'être directement intégré au processus de chaînage du contour, ce qui permet de décroître le temps calcul. D'autre part, il est possible d'utiliser la même approche en introduisant d'autres critères de distances comme, par exemple, le maximum de courbure.

## 4 Estimation récursive le long de la séquence

Du fait des perturbations, des erreurs de modélisation, des bruits de mesure (discrétisation, quantification), les primitives 3D reconstruites à partir de seulement un couple d'images sont souvent très bruitées. Une idée intéressante consiste à utiliser la cohérence temporelle pour affiner l'estimée au long de la séquence. Formuler le problème en termes de filtrage récursif devient alors très intéressant.

### 4.1 Cas des primitives "points"

#### 4.1.1 Cas général

Le vecteur d'état est constitué par les coordonnées 3D des points dans le repère  $R_s$ , et l'équation d'état (eq. 9) peut être réécrite sous la forme :

$$\dot{\underline{x}} = \underline{V}(t) + \omega(t)\underline{x} \quad (74)$$

où  $\omega(t)$  est la matrice antisymétrique associée au produit vectoriel.



Les équations de mesures proviennent des équations (eqs. 4 et 16) et prennent la forme

$$\begin{cases} \hat{x} = \frac{X}{Z} + e_x(t) \\ \hat{y} = \frac{Y}{Z} + e_y(t) \end{cases} \quad (75)$$

$$\quad (76)$$

$$\hat{v}_\perp = v_\perp + e_v(t) = \lambda_1(u_T + v_T) + \lambda_2(u_F + v_F) + e_v(t) \quad (77)$$

avec

$$\begin{cases} u_T = \frac{1}{Z} (-v_X + \frac{X}{Z} v_Z) \end{cases} \quad (78)$$

$$\begin{cases} u_F = \frac{1}{Z^2} (YZ\Omega_Z + XY\Omega_X - (X^2 + Z^2)\Omega_Y) \end{cases} \quad (79)$$

$$\begin{cases} v_T = \frac{1}{Z} (-v_Y + \frac{Y}{Z} v_Z) \end{cases} \quad (80)$$

$$\begin{cases} v_F = \frac{1}{Z^2} (-XZ\Omega_Z - XY\Omega_Y + (Y^2 + Z^2)\Omega_X) \end{cases} \quad (81)$$

Plusieurs problèmes apparaissent à ce niveau concernant :

\* **les perturbations** : Le bruit d'état est du à l'erreur de mesure sur le mouvement, malgré qu'il soit de la forme suivante :

$$\begin{cases} \hat{v}(t) = v(t) + e_v(t) \end{cases} \quad (82)$$

$$\begin{cases} \hat{\Omega}(t) = \Omega(t) + e_\omega(t) \end{cases} \quad (83)$$

cela entraîne un bruit multiplicatif au niveau de l'équation (eq. 74).

De plus, les bruits sur les équations de mesure apparaissent également de façon non linéaire:  $\hat{\lambda}_1$  et  $\hat{\lambda}_2$  dépendent de l'estimation de l'orientation du contour fournie par l'EOF qui est bruitée.

\* **la discrétisation** : Le passage du modèle continu au modèle discret utilisé en pratique pose des problèmes au niveau des rotations : choix de la représentation des orientations, transformation des équations (eqs. 79 et 81) en termes de vecteurs de rotations discrets.

\* **les nonlinéarités** : Une autre difficulté importante vient du fait que toutes les équations de mesures sont non linéaires vis à vis des variables d'états :

$$\underline{z} = \underline{f}(\underline{X}, \underline{V}, \underline{\Omega}) + \underline{e}(t) \quad (84)$$

La technique largement utilisée pour contourner cette difficulté consiste à linéariser à chaque instant les équations autour de la valeur  $\underline{X}^*(t)$  la plus proche possible de la valeur réelle, soit :

$$\underline{z} \approx \underline{f}(\underline{X}^*(t), \underline{V}, \underline{\Omega}) + \frac{\partial \underline{f}}{\partial \underline{X}}(\underline{X}^*(t), \underline{V}, \underline{\Omega}) (\underline{X} - \underline{X}^*(t)) + \dots \quad (85)$$

Dans le cas discret, l'algorithme le plus classique est connue sous le nom de *filtrage de Kalman étendu*, où  $\underline{X}^*(t)$  est la *prédiction* de  $\underline{X}(t)$  au temps  $t = t_k$ . Alors, les méthodes classiques de filtrage peuvent être employées sur l'équation (eq. 84). Cependant, une telle approche peut entraîner des problèmes au niveau de la convergence.

Pour toutes ces raisons, les algorithmes d'estimation récursive seront généralement approximatés. Dans le prochain paragraphe, nous étudierons le cas plus simple d'un capteur se déplaçant uniquement en translation.

#### 4.1.2 Mouvement de translation

Le passage en discret est direct dans ce cas et nous pouvons écrire, pour chaque instant  $k$  et pour chaque point  $X$ , l'équation d'état :

$$\underline{X}(k+1) = \underline{X}(k) + \underline{V}(k) + \underline{\varepsilon}_V(k+1) \quad (86)$$

où nous supposons que  $\varepsilon_V$  est un bruit blanc centré de variance  $q_V I$ .

En addition aux équations (eqs. 75 et 76), l'équation principale de mesure est fournie par l'EOF (eq. 69), et il est impossible d'éviter les problèmes dus aux non linéarités et aux bruits. En vue de rendre possible les calculs, plusieurs approximations sont requises.

L'équation (eq. 69) dans le cas d'un mouvement de translation, s'écrit :

$$\tilde{\underline{z}} = \frac{1}{Z} \left( -(\lambda_1^{\tilde{}} \tilde{\underline{V}}_X + \lambda_2^{\tilde{}} \tilde{\underline{V}}_Y) + (\lambda_1^{\tilde{}} \frac{X}{Z} + \lambda_2^{\tilde{}} \frac{Y}{Z}) \tilde{\underline{V}}_Z \right) + \underline{e}_3 \quad (87)$$

où le "tilde" représente le caractère bruité de la variable. Une première approximation consiste à considérer une absence de bruit sur les équations (eqs. 75 et 76). Cela permet de remplacer  $\frac{X}{Z}$  et  $\frac{Y}{Z}$  dans (eq. 87) par les valeurs "exactes"  $x$  et  $y$ , et donc de réduire la dimension du problème à l'estimation unique de  $Z$ . La seconde approximation est destinée à linéariser l'équation de mesure, nous proposons deux façon de réaliser cela :

- Soit linéariser (eq. 8 ) autour d'un point donné  $\underline{Z}^*$ , on obtient alors :

$$\begin{aligned} \tilde{V}_Z &\approx \frac{a}{Z^*Z} (ZZ^* - Z) + e_v \\ a &= -(\lambda_1^* \tilde{V}_X + \lambda_2^* \tilde{V}_Y) + (\lambda_1^* x + \lambda_2^* y) \tilde{V}_Z \end{aligned} \quad (88)$$

- Soit appliquer directement l'équation de mesures

$$\tilde{Z} = \frac{a}{\tilde{V}_Z} Z + e_Z \quad (89)$$

Dans les deux cas  $e_v$  et  $e_Z$  sont des termes globaux de bruit prenant en compte les contributions dues à la linéarisation et aux approximations. Il est nécessaire de considérer que leur covariance n'est pas constante (elle dépend même de l'état), et, en pratique, de la borner. Dans la suite, nous traiterons le cas de l'approximation (eq. 89) et les équations seront alors:

$$\begin{cases} Z(k+1) = Z(k) + V_Z(k) + \varepsilon_Z(k+1) \end{cases} \quad (90)$$

$$\begin{cases} \tilde{Z}(k+1) = Z(k+1) + e_Z(k+1) \end{cases} \quad (91)$$

où  $\tilde{Z}$  est calculé à chaque instant en utilisant (eq. 89), et en supposant que:  $\varepsilon_Z$  est centré, indépendant de variance  $q$  et  $e_Z(k)$  est centré, indépendant de variance  $r(k)$  où  $k$  est le numéro de l'image dans la séquence.

Les équations de filtrage associées sont les équations bien connues du filtre de Kalman, dans lesquelles on peut introduire, si nécessaire, un facteur d'oubli exponentiel. La forme générale de l'équation d'estimation pour  $Z$  est alors :

$$\hat{Z}(k+1) = \hat{Z}(k) + V_Z(k) + K(k+1) (\tilde{Z}(k+1) - V_Z(k) - \hat{Z}(k)) \quad (92)$$

où  $K(k+1)$  est le gain de Kalman.

**Remarque :** L'approche présentée ci-dessus concerne uniquement un même point suivi au long de la séquence, nous montrerons dans la section suivante comment l'étendre à un ensemble de points.

## 4.2 Cas des primitives "segments"

Nous réduirons l'étude au cas des translations, pour un segment de droite sous la contrainte donnée par (eq. 35). L'état du système est fourni par les versions discrètes des équations (eqs. 31 et 37), avec  $\Omega = 0$ . Cela nous procure quatre équations dont deux sont linéaires et deux non linéaires avec le vecteur d'état suivant :

$$R(k) = (\alpha(k) \beta(k) h_1(k) h_2(k))^T \quad (93)$$

Les variables de mesures à chaque instant sont  $p(k)$  et  $q(k)$ . Les équations de mesures (eqs. 33 et 34) sont nonlinéaires et il est nécessaire de les linéariser avant d'appliquer le filtre récursif. Cependant  $p(k)$  et  $q(k)$  sont évalués grâce à une estimation spatiale. Si le segment est assez long et si la dispersion des points le constituant n'est pas très importante, l'erreur résultante sur le segment sera plus faible que dans le cas de l'estimation des points. Aussi, l'utilisation directe des équations (eqs. 63 et 64) en tant qu'équations de mesure est possible. L'algorithme pourra alors être de la forme :

i) A partir des mesures  $\tilde{\alpha}(k)$  et de  $\tilde{\beta}(k)$  fournies par (eqs. 63 et 64), mettre à jour l'état avec un filtre linéaire portant sur l'équation :

$$\begin{pmatrix} \alpha(k+1) \\ \beta(k+1) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha(k) \\ \beta(k) \end{pmatrix} + \text{noise} \quad (94)$$

ii) Partant des estimées courantes  $\hat{\alpha}(k)$  et  $\hat{\beta}(k)$ , calculer  $\hat{\gamma}(k)$

iii) Mettre à jour  $(\hat{h}_1(k), \hat{h}_2(k))$  en utilisant les mesures  $(\tilde{h}_1(k), \tilde{h}_2(k))$  et les équations d'état approximées :

$$\begin{pmatrix} h_1(k+1) \\ h_2(k+1) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} h_1(k) \\ h_2(k) \end{pmatrix} + \hat{\gamma}(k) \begin{pmatrix} v_Y(k) \\ -v_X(k) \end{pmatrix} + v_Z(k) \begin{pmatrix} -\hat{\beta}(k) \\ \hat{\alpha}(k) \end{pmatrix} \quad (95)$$

## 5 Elaboration de la commande en boucle fermée

Une constatation évidente est que la qualité de l'estimation récursive au long de la séquence d'images est très sensible aux mouvements successifs du capteur (amplitude et direction). De façon plus précise, la nature de ce mouvement affecte les erreurs de mesure dans l'image : par exemple, le mouvement du capteur qui produira le glissement d'un segment de l'image sur lui-même n'induera aucune information sur la structure du segment 3D correspondant dans la scène.

Dans le cas des robots mobiles, il est possible de contrôler le mouvement de la caméra (bien que la mesure de ce mouvement soit bruitée). Nous allons utiliser cette possibilité pour réaliser un schéma de commande en boucle fermée dans lequel nous allons superposer au mouvement "nominal" requis par la tâche, un mouvement destiné à accroître la qualité des estimations. Cette idée est, en un certain sens, proche du concept d'*extra-signal* utilisé dans certaines méthodes d'identification. Nous allons décrire cette approche dans le cas des "points" et des "segments".

### 5.1 Cas des points

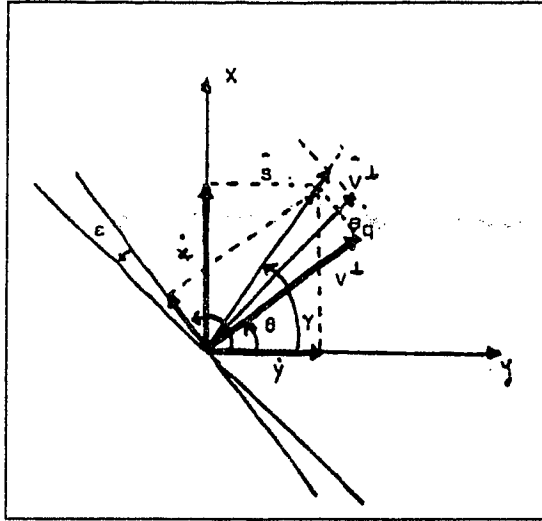


Figure 3. Erreurs de mesures introduites par l'EOF

La figure 3 représente la modélisation des erreurs de mesures introduites par l'EOF, soit :

$$\tilde{\theta} = \theta + \epsilon_{\theta} \quad (96)$$

$$\tilde{v}_{\perp} = \frac{v_{\perp}}{\cos \epsilon_{\theta}} + \epsilon_v \quad (97)$$

où  $\epsilon_{\theta}$  correspond au bruit de discrétisation sur l'orientation du contour et  $\epsilon_v$  le bruit de quantification sur  $\|\tilde{v}_{\perp}\|$ .

Dans la pratique,  $\epsilon_{\theta}$  est uniquement fonction du jeu de masques utilisés dans l'EOF, il peut être borné et choisi petit :

$$\cos \epsilon_{\theta} \simeq 1 \quad (98)$$

$$\sin \epsilon_{\theta} \simeq \epsilon_{\theta} \quad (99)$$

ce qui a pour conséquence que :

$$\tilde{\lambda}_1 = \lambda_1 - \lambda_2 \epsilon_{\theta} \quad (100)$$

$$\tilde{\lambda}_2 = \lambda_2 + \lambda_1 \epsilon_{\theta} \quad (101)$$

Posons les autres erreurs de mesures (mouvement et coordonnées pixel) égale à :

$$\tilde{V} = V + \varepsilon_V \quad (102)$$

$$\tilde{x} = x + e_x \quad (103)$$

$$\tilde{y} = y + e_y \quad (104)$$

alors, l'approximation de l'erreur relative sur  $\tilde{Z}$  (eq. 89) prend la forme :

$$\frac{\tilde{Z} - Z}{Z} \approx \varepsilon_\theta \frac{V^T}{V^I} - \frac{\varepsilon_V}{V^I} + \frac{h'}{ZV^I} + 2^{\text{nd}} \text{ order terms} + \dots \quad (105)$$

avec

$$h' = -(\varepsilon_X \lambda_2 + \varepsilon_Y \lambda_1) + \varepsilon_Z(\lambda_1 y + \lambda_2 x) + V_Z(\lambda_1 \varepsilon_X + \lambda_2 \varepsilon_Y) \quad (106)$$

Cette erreur sera réduite si :

- $\varepsilon_\theta$  est petit : Ce bruit est lié uniquement à l'EOF, il ne dépend pas du mouvement et peut être réduit.
- $h'$  est petit : Cela nécessite de minimiser  $\varepsilon_V$ , donc d'avoir la meilleure mesure possible du mouvement. Du point de vue de la commande en boucle fermée, il faut également que  $V_Z$  soit le plus petit possible.
- $\frac{1}{V^I}$  et  $\frac{V^T}{V^I}$  sont petits : Cette condition traduit que localement, le vecteur d'optic flow doit être orthogonal au contour avec une amplitude maximum.

En tenant compte de ces contraintes et en gardant à l'esprit le fait que nous devons rester le plus proche possible de la trajectoire nominale  $V^N(k)$  imposée par la tâche, nous allons poser le problème en termes de minimisation.

Le EFO est calculé entre l'image  $(k-1)$  et l'image  $(k)$ , le but est de trouver le mouvement "optimal" du capteur entraînant la meilleure mesure d'optic flow entre  $(k)$  et  $(k+1)$ . Nous supposons le suivi des points au long de la séquence résolu et que chacun d'eux correspond à un même point 3D dans la scène :

$$\underline{p}_i(k-1) = (x_i(k-1) \ y_i(k-1))^T \quad (107)$$

$$i = 1 \dots N(k-1)$$

A chaque point  $p_i(k-1)$  nous associons

- \* une position estimée  $\tilde{p}_i(k)$  dans l'image  $k$
- \* la mesure fournie par l'EOF (eqs. 96 et 97)

$$\left\{ \frac{\tilde{V}_i}{V^I} (k-1) \ ; \ \tilde{\theta}_i(k-1) \right\}$$

- \* un déplacement local désiré de  $p_i(k)$  entre l'image  $k$  et l'image  $k + 1$  appelé par la suite  $v_d^i(k)$
- \* un paramètre de pondération  $\sigma^i(k)$

Posons :

$$\underline{E}(k) = (\underline{v}_d(k) - \underline{\tilde{v}}^1(k)) \dots (\underline{v}_d^{N(k)} - \underline{\tilde{v}}^N(k)) \quad (108)$$

où  $\tilde{v}^i(k)$  est la prédiction du déplacement local de  $\tilde{p}^i(k)$  de l'image  $k$  à l'image  $k + 1$  (calculée à partir de  $\tilde{x}_i(k), \tilde{y}_i(k)$  et de  $\tilde{Z}_i(k)$  l'estimation de  $Z_i(k)$  fournie par le filtre récursif décrit précédemment), qui dépend explicitement de  $V(k)$ , le mouvement du capteur entre  $k$  et  $k + 1$  qui doit être optimiser.

Notons

$$W(k) = \begin{pmatrix} \sigma_{[k]}^1 & & 0 \\ & \ddots & \\ 0 & & \sigma_{[k]}^{N(k)} \end{pmatrix} \quad (109)$$

, la fonction de coût à minimiser sera finalement :

$$J(k) = \lambda(k) \underline{E}^T(k) W(k) \underline{E}(k) + (1-\lambda(k)) \|\underline{V}(k) - \underline{V}^N(k)\|^2 + \xi V_Z^2 \quad (110)$$

dont la solution sera donnée par :

$$\frac{\partial J(k)}{\partial V(k)} (V^*(k)) = 0 \quad (111)$$

Cette méthode sera appliquée à un problème d'évitement d'obstacles dans la section suivante.

## 5.2 Cas des segments

L'estimation des paramètres de type "segments" présente quelques différences par rapport la méthode précédente : en fait la première étape (correspondant à l'EOF) est le calcul des paramètres  $(\hat{p}, \hat{q})$ , qui est obtenu par approximation polygonale.

Cette étape est moins locale que l'EOF, de ce fait on peut espérer que les estimations obtenues seront moins bruitées que dans le cas de points. C'est pourquoi le raffinement de l'estimée au long de la séquence peut amener à de bons résultats sans nécessiter de s'écarter de la trajectoire nominale. Toutefois, si cela s'avère nécessaire, la même démarche que pour les points pourra être mise en oeuvre.

L'analyse du "bruit" sur les équations de mesures (eqs. 63 et 64) ne sera pas détaillée ici. Elle montre que la meilleure qualité d'estimation, en termes de S/B, est obtenue lorsque le mouvement du capteur est tel que, entre l'image  $k$  et l'image  $k + 1$ ,  $p(k + 1) - p(k)$  est grand et  $q(k + 1) - q(k)$  est petit. En appelant les valeurs désirées de ces variations  $\Delta^* p(k + 1), \Delta^* q(k + 1)$ , nous pouvons construire la fonction de coût de la même façon que précédemment :

$$\begin{aligned}
 J'(k) = & \lambda(k) \sum_{j=1}^{m(k)} \{ \sigma_j^j(k) (\tilde{p}_j(k+1) - \tilde{p}_j(k) - \Delta^* p(k+1))^2 \\
 & + \tau_j^j(k) (\tilde{q}_j(k+1) - \tilde{q}_j(k) - \Delta^* q(k+1))^2 \\
 & + (1 - \lambda(k)) \| \underline{v}(k) - \underline{v}^N(k) \|^2
 \end{aligned} \tag{112}$$

pour chaque ligne  $L_j, j = 1 \dots m(k)$ .

En utilisant les équations (eqs. 33,34,94 et 95), et les résultats du filtre récursif, on obtient une expression approximée de  $J'(k)$  qui dépend explicitement de  $V(k)$ . Cependant, du fait des non linéarités, une méthode numérique est nécessaire pour calculer  $V^*(k)$  minimisant  $J'(k)$ .

## 6 Application en robotique

Dans cette dernière section, nous allons présenter deux applications dans le domaine de la robotique utilisant l'approche décrite précédemment. Les résultats ont été obtenus sur un outil de simulation développé IRISA et qui permet de simuler les images perçues par une caméra mobile se déplaçant dans un univers 3D au moyen de techniques de synthèse d'image [22].

Nous allons tout d'abord décrire une simple application d'évitement d'obstacles, puis nous superposerons un mouvement destiné à renforcer l'estimation suivant l'approche décrite dans la section IV. Ces deux expérimentations utilisent des primitives 3D de type "points", l'estimation des primitives de type "segments" n'étant pas encore implantée.



## 6.1 Commande simple en boucle fermée

Dans ce premier exemple, l'estimation récursive n'est pas réalisée et les coordonnées 3D calculées à partir des équations (eqs. 82, 4 et 5) sont très bruitées. Considérons la tâche locale d'évitement d'obstacles suivante : partant d'une position initiale dans le repère  $R_s$ , nous désirons atteindre un but  $X^g$  en évitant les obstacles. La trajectoire nominale est donnée par:

$$\underline{V}^M = \eta \underline{X}_c^g \quad (113)$$

où l'indice  $c$  représente l'expression des coordonnées du but dans le repère de la caméra.  $\eta$  peut avoir la forme suivante :

$$\eta = \underline{V}_0 \parallel \underline{X}_c^g \parallel^{-1} \quad (114)$$

La tâche d'évitement d'obstacles est réalisée en ajoutant à  $\underline{V}^M$  un mouvement local  $\underline{V}_i$  calculé en associant aux obstacles  $\{O\}$  un champ de potentiel de type Newtonnien, tel que :

$$\forall \underline{P}_i (x^i, y^i, z^i) \in \{O\} \quad \underline{V}_i^1 = -w_i \frac{\overrightarrow{0_c P_i}}{\parallel \overrightarrow{0_c P_i} \parallel^3} \quad (115)$$

exprimé dans  $R_c$ , avec un coefficient de pondération  $w_i$

Naturellement,  $\underline{V}_i$  est calculé à partir de la valeur estimée de  $\tilde{Z}_i$ , ce qui finalement donne à chaque instant :

$$\underline{V}_i^1(k) = \frac{1}{N(k)} \sum_{i=1}^{N(k)} \frac{w_i}{\tilde{Z}_i^2(k) (\tilde{x}_i(k) + \tilde{y}_i(k) + 1)^{3/2}} \underline{x}^P \quad (116)$$

Le mouvement résultant est alors :

$$\underline{V}^N(k) = \underline{V}^M(k) + \underline{V}_e(k) \quad (117)$$

La figure suivante [16] montre un exemple de trajectoire générées par cet algorithme.

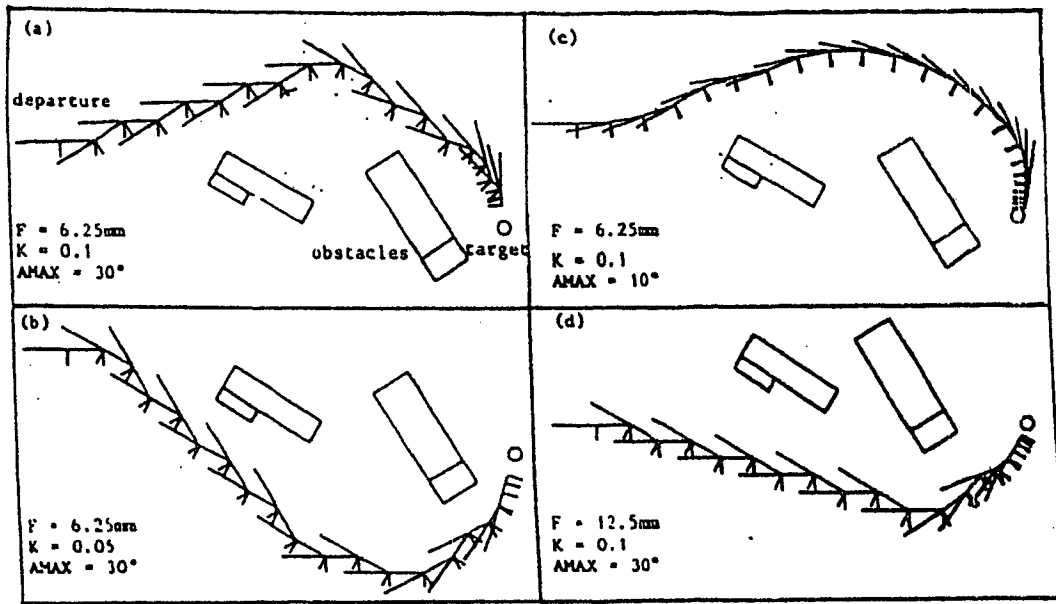


Figure 4. Exemples de trajectoires générées par l'algorithme d'évitement d'obstacles

Le principal intérêt de ce simple exemple est de montrer que, dans une approche en boucle fermée, le processus de lissage peut être situé en dehors du processus d'estimation et de mesure, quand la tâche le permet; ici l'étape de "moyennage" des mesures locales est intégrée dans le calcul du déplacement global.

## 6.2 Estimation récursive avec commande optimale

Dans cette seconde expérimentation, l'approche décrite dans le rapport a été totalement implantée. La scène est du même type que précédemment et le mouvement nominal est défini par la même tâche: atteindre le but en évitant localement les obstacles.  $V^N(k)$  est alors défini par l'équation (eq. 117), avec  $\hat{Z}_i(k)$  fourni par le filtre récursif. La résolution complète de l'équation (eq. 110) est donnée en Annexe B. Les différents paramètres ou variables qui apparaissent dans l'algorithme sont les suivants:

a. Le mouvement désiré  $v_d^i(k)$  :

Il est défini par (cf. section IV)

$$\begin{cases} v_d^{iT} = 0 \\ v_d^{i\perp} = v_d^i \end{cases} \quad (118)$$

$$(119)$$

$v_d^i$  est la valeur maximale autorisée par l'EOF

Nous pouvons écrire alors :

$$\underline{v}_d^1(k) = \begin{pmatrix} \tilde{\beta}_1(k) & \tilde{\alpha}_1(k) \\ \tilde{\alpha}_1(k) & -\tilde{\beta}_1(k) \end{pmatrix} \begin{pmatrix} v_d^1 \\ 0 \end{pmatrix} \quad (120)$$

et

$$\underline{v}_d^1(k) = \begin{pmatrix} \tilde{\beta}_1(k) \\ \tilde{\alpha}_1(k) \end{pmatrix} v_d^1$$

b. Le mouvement prédit  $\tilde{v}^i(k)$  :

Il est donné par :

$$\underline{\tilde{v}}^1(k) = \frac{1}{\tilde{z}_1(k)} \begin{pmatrix} -V_X(k) + \tilde{x}_1(k) V_Z(k) \\ -V_Y(k) + \tilde{y}_1(k) V_Z(k) \end{pmatrix} \quad (121)$$

c. Les coordonnées pixel estimées dans l'image  $k$  :

$$\begin{pmatrix} \tilde{x}_1(k) \\ \tilde{y}_1(k) \end{pmatrix} = \begin{pmatrix} x_1(k-1) \\ y_1(k-1) \end{pmatrix} + \underline{\tilde{v}}_1(k-1) \quad (122)$$

$\tilde{v}_1(k-1)$  est calculé à partir de l'EOF, et défini par sa pente dans le repère  $R_c$  (angle  $\delta_1(k-1)$ ) et sa norme :

$$\text{tg } \delta_1(k-1) = \frac{-V_Y^*(k-1) + y_1(k-1) V_Z^*(k-1)}{-V_X^*(k-1) + x_1(k-1) V_Z^*(k-1)} \quad (123)$$

$$\| \underline{\tilde{v}}_1(k-1) \| = \frac{\tilde{v}_1^{\perp}(k-1)}{\cos(\delta_1(k-1) - \tilde{\theta}_1(k-1))} \quad (124)$$

d. Le paramètre de pondération locale  $\sigma^i(k)$  :

Ces paramètres sont un point clé de la méthode. L'idée de base provient du fait que dans une image la direction des contours est aléatoire et, donc, les mouvements locaux désirés également (orthogonaux aux contours) il en résultera que le mouvement global "optimal" découlant de la minimisation du critère sera oscillant ou n'aura aucun sens. C'est pourquoi, nous associons à chaque point, un paramètre  $\sigma'_i(k)$  mesurant la qualité de l'estimation courante. Ainsi, à chaque instant, le mouvement superposé favorisera les points les plus mal estimés. Nous avons testé deux types de fonctions :

$$\sigma'_i(k) = q_i(k) \quad (125)$$

et

$$\sigma'_i(k) = \frac{q_i(k)}{z_i(k)} \quad (126)$$

avec

$$\sigma_i(k) = \frac{\sigma'_i(k)}{\sum_{i=1}^{N(k)} \sigma'_i(k)} \quad (127)$$

$q_i(k)$  est la variance de l'erreur d'estimation associée au filtre pour chaque point et à chaque instant. L'introduction de la profondeur estimée dans (eq. 126) a deux avantages :

- Du point de vue de la tâche, les points les plus proches sont les plus importants: leur poids est augmenté dans le calcul du mouvement optimal.
- Obtenir la valeur désirée  $v_d$  pour des points éloignés suppose des mouvements de la caméra de grande amplitude ce qui perturbe beaucoup la trajectoire nominale et rend plus difficile l'estimation des points les plus proches.

#### e. Le paramètre de pondération globale $\lambda(k)$ :

Le but de ce paramètre est d'éviter des perturbations inutiles de la tâche nominale lorsque les points sont bien estimés. Nous définissons un index global de qualité:

$$I(k) = \frac{1}{N(k)} \sum_{i=1}^{N(k)} q_i(k) \quad (128)$$

et nous choisissons, par exemple :

$$\lambda(k) = 1 - e^{-\partial I(k)} \quad (129)$$

Le dernier point important dans la mise en oeuvre de cet algorithme concerne la mise en correspondance locale des points: il est nécessaire que les mesures données par l'EOF pour

un point  $i$  tout au long de la séquence correspondent au même point 3D dans la scène. Deux problèmes doivent être résolus :

- **mise en correspondance** : Une fenêtre 5X5 est centrée sur la valeur prédite  $\tilde{x}_i(k+1), \tilde{y}_i(k+1)$ , et une fonction de distance entre ce point et les autres points appartenant au contour contenu dans la fenêtre est calculée. Le plus proche voisin est alors mis en correspondance avec le point prédit.
- **gestion des entrées/sorties** : Les points entrants dans l'image en cours de séquence sont initialisés avec la profondeur moyenne dans la scène et une grande variance leur est associée. Les points quittant l'image sont détectés lorsque leur prédiction sort du domaine de l'image.

La figure 5 résume l'ensemble de l'algorithme et la figure 6 montre la trajectoire obtenue dans un cas similaire à celui de la première expérimentation.

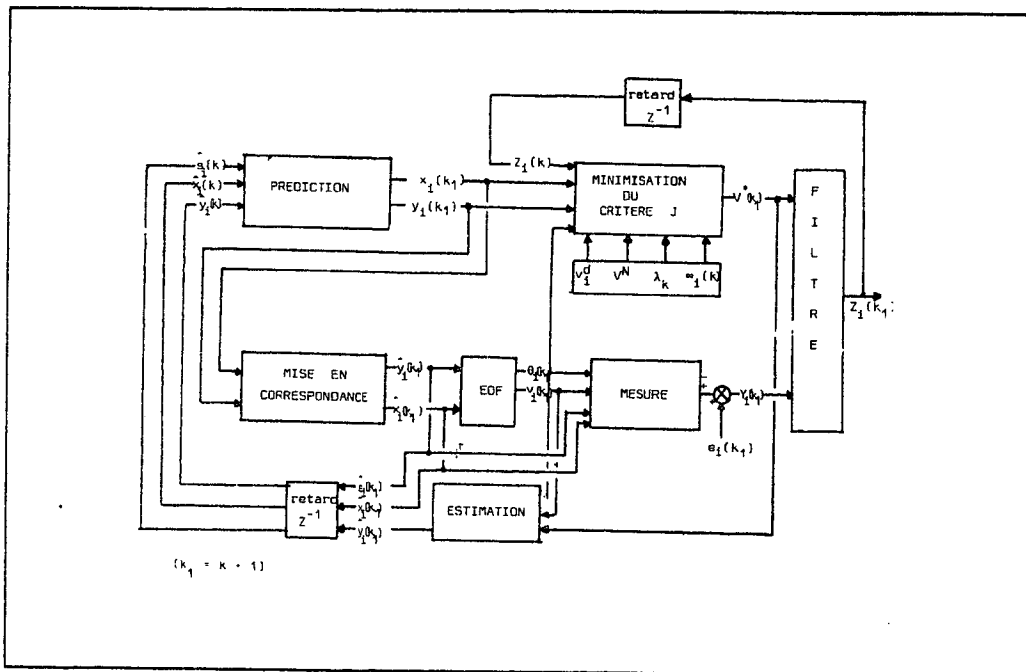


Figure 5. Schéma fonctionnel de l'algorithme

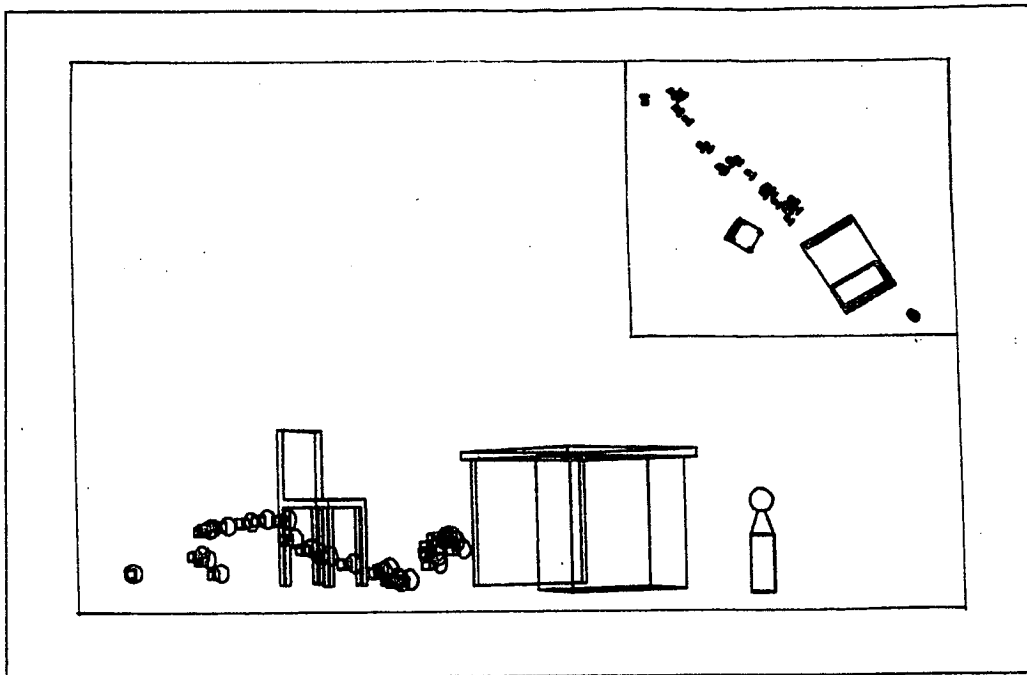


Figure 6. Exemple de trajectoire avec estimation récursive

## 7 Conclusion

Nous avons présenté dans ce rapport un schéma d'estimation récursive de primitives 3D utilisant des informations dynamiques extraites d'une séquence d'images. Une méthode originale de contrôle en boucle fermée a été implantée dans le cas de l'estimation de la carte des profondeurs avec une caméra se déplaçant en translation. Ces méthodes ont été validées en simulation.

Cependant, de nombreuses améliorations restent à faire dans le domaine de l'estimation récursive de primitives 3D au moyen du mouvement.

En premier, concernant les algorithmes décrits, des efforts doivent être fait pour simplifier les modèles et décroître les coûts calcul; la validation sur images réelles est également nécessaire.

Concernant les primitives de type "segments", le schéma d'estimation proposé doit être vérifié et le passage à des primitives de type "surfaces" reste à faire. Finalement, il faut également remarquer que l'extension de l'approche au cas des rotations n'est pas directe et nécessite des précautions notamment au niveau du choix des représentations.

En conclusion, nous présentons que des tâches complexes en univers inconnu peuvent être réalisées par un robot avec l'approche proposée qui permet d'acquérir *en ligne*, durant le déplacement et avec une caméra unique, les informations nécessaires à la bonne exécution de la tâche. Ce moyen pourrait être un bon complément ou même une alternative aux méthodes classiques de stéréovision ou de vision 3D active.

## REFERENCES

- [1] AGGARWAL, J.K.; DAVIS, L.S.; MARTIN, W.N.: Correspondence processes in dynamic scene analysis. **Proc. of IEEE**, vol.69, n°5, 1981.
- [2] BERTHOD, M.; LONG, P. : Stéréoscopie par mise en correspondance de graphes. **Colloque CESTA/GRETSI**, Biarritz, France, May, 1984.
- [3] YAKIMOVSKY, Y.; CUNNINGHAM, R.: "A system for extracting Three-dimensional Measurements from a stereo pair of T.V. Cameras". **Computer Graphics and Image Processing**, vol.7, p.195-210, 1978.
- [4] AYACHE, N.; FAVERJON, B.: Fast stereo matching of edge segments. **IEEE Conf. on CVPR**, San Francisco, June 1985.
- [5] BUXTON, R.F.; BUXTON, H. : Monocular depth perception from optical flow by space time signal processing. **Proc. R. Soc.**, London, B218, 1983.
- [6] HILDRETH, E.C. : The measurement of visual motion. **ACM distinguished dissertation**, MIT Press, 1983.
- [7] HILDRETH, E.C. : Computations underlying the measurement of visual motion. **Artificial Intelligence**, n°23, 1984.
- [8] BOUTHEMY, P. : Estimation of edge motion based on local modelling. **SPIE Conference on Robot Vision**, Cannes, France, Dec. 1985.
- [9] NAGEL, H.H. : Recent advances in image sequence analysis. **Colloque CESTA/GRETSI**, Biarritz, France, May 1984.
- [10] BRUSS, A.; HORN, B.K.P. : Passive navigation. **CVG & IP** n°21, 1983.
- [11] TSAI, R.Y.; HUANG, T.S.: "Uniqueness and Estimation of Three-dimensional Motion parameters of rigid objects with curved surfaces". **IEEE Trans. on PAMI**, vol.PAMI-6, n°1, Janv. 1984.

- [12] WEBB, J.; AGGARWAL, J.K.: "Structure from motion of rigid and jointed objects". **Artificial Intelligence**, vol.19, p.107-130, 1982.
- [13] MORAVEC, H.P. : Robot visual obstacle avoidance. **IJCAI**, 1981.
- [14] RIEGER, J.H. : Information in optical flow induced by curved path of observation. **Journal of Opt. Soc. Am.**, vol.73, n°3, March 1983.
- [15] SANDERSON, A.C.; WEISS, L.E. : Adaptive visual servo-control of robots. **Robot vision**, ed. A. Pugh, 1982.
- [16] RIVES, P.; MARCE, I.: Use of moving vision sensors in robotics. Application to an obstacle avoidance task. **ICAR '85**, Tokyo, Sept. 1985.
- [17] NEVATIA, R.; BABU, R. : Line feature extraction and description. **CGIP** 13, p.257-269, 1983.
- [18] GRIMSON, W.E.L.; HILDRETH, E.C. : Comments on "Digital step edges from zero crossings of second directional derivatives". **IEEE Trans. PAMI**, vol. PAMI 7, pp.121-129, Jan. 1985.
- [19] DERICHE, R. : Optimal edge detection using recursive filtering. Submitted to **International Journal on Computer Vision**.
- [20] CANNY, J.F. : Finding edges and lines in images. **MIT Artificial Intelligence laboratory**, TR-720, 1983.
- [21] BALLARD, D.H.: Strip trees: a hierarchical representation for curves. **Com. of ACM**, vol.24, n°5, p.310-321. May, 1981.
- [22] RIVES, P.; HEGRON, G. : Design of a simulation tool for robots using moving vision sensors. **Proc. of NATO Int. Adv. Res. workshop on languages for sensor-based control in robotics**. Castelvechio Pascoli, Italy, Sept. 1986.



## ANNEXE A

## Principales notations

**k: index temporel dans la séquence**  
**i : index spatial dans l'image**

**\*Paramètres 3D**

$R_S$	Repère scène
$R_C$	Repère image
$\underline{X} = (X \ Y \ Z)^T$	Coordonnées 3D dans $R_C$
$\hat{Z}(k)$	Estimation de Z à l'instant k
$q(k)$	Variance de l'erreur d'estimation sur $\hat{Z}(k)$
$\hat{Z}(k)$	Mesure de Z à l'instant k
$\varepsilon_Z$	Bruit sur $\hat{Z}$ après linéarisation
$\underline{V} = (V_X \ V_Y \ V_Z)^T$	Vitesse en translation de $R_C$
$\underline{\Omega} = (\Omega_X \ \Omega_Y \ \Omega_Z)^T$	Vitesse en rotation de $R_C$
$\underline{\varepsilon}_V = (\varepsilon_X \ \varepsilon_Y \ \varepsilon_Z)^T$	Bruit sur $\underline{V}$
$q_V$	Variance de $\underline{\varepsilon}_V$
$\underline{\varepsilon}_\Omega$	Bruit sur $\underline{\Omega}$
$\underline{\hat{V}} = \underline{V} + \underline{\varepsilon}_V$	
$\underline{\hat{\Omega}} = \underline{\Omega} + \underline{\varepsilon}_\Omega$	
$\underline{V}^N$	Mouvement nominal
$\underline{V}^M$	Mouvement d'attraction de la cible
$\underline{V}^e$	Mouvement de répulsion des obstacles
$\underline{V}^*$	Mouvement optimal calculé
$\underline{X}_0$	Coordonnées 3D d'un point appartenant à $\mathcal{D}$
$\underline{u} = (\alpha \ \beta \ \gamma)^T$	Vecteur directeur de $\mathcal{D}$
$\underline{h} = (h_1 \ h_2 \ h_3)^T = \underline{X}_0 \times \underline{u}$	

**\*Paramètres 2D**

$$\left. \begin{aligned} p &= \frac{h_1}{h_2} \\ q &= \frac{\alpha p - \beta}{\gamma} \end{aligned} \right\}$$

Paramètres du segment projeté dans l'image

$$\underline{x}^P = (x \ y \ 1)$$

Coordonnées de la projection p de P dans l'image

$$e_x, e_y$$

Bruits sur x,y

$$\underline{v} = (\dot{x} \ \dot{y})^T = (v^\perp \ v^T)^T$$

Mouvement de p

$$v^\perp$$

Composante orthogonale au contour de  $\underline{v}$ 

$$v^T$$

Composante tangentielle de  $\underline{v}$ 

$$u_T, v_T$$

Vitesse en translation de p

$$u_r, v_r$$

Vitesse en rotation de p

$$\theta$$

Orientation du contour spatial

$$v^*$$

Mesure fournie par l'EOF

$$\tilde{v}^\perp = v^\perp + \varepsilon_v$$

Mesure de  $v^\perp$  fournie par l'EOF

$$\tilde{\theta} = \theta + \varepsilon_\theta$$

Mesure de  $\theta$  fournie par l'EOF

$$\varepsilon_\theta$$

Bruit de quantification sur  $\theta$ 

$$\varepsilon_v$$

Bruit de quantification sur  $v$ 

$$\lambda_1^v = \cos \tilde{\theta}$$

$$\lambda_2^v = \sin \tilde{\theta}$$

$$\underline{z}$$

Equation générale de mesure

$$\underline{\tilde{z}}$$

Equation générale de mesure linéarisée

$$e_v$$

Bruit sur  $\tilde{v}^\perp$  après linéarisation

$$\left( \tilde{x}_i(k), \tilde{y}_i(k) \right)$$

Position estimée dans l'image k du point i

$\hat{v}_i(k-1)$	Estimation du vecteur déplacement entre $k-1$ et $k$
$\hat{\delta}_i(k-1)$	Pente de $\hat{v}_i(k-1)$
$\hat{v}_i(k)$	Prédiction du mouvement de $\hat{p}_i(k)$ dans l'image $(k+1)$
$\underline{v}_i^1(k)$	Mouvement désiré dans l'image
$\sigma_i(k)$	Paramètre de pondération locale
$\lambda(k)$	Paramètre de pondération globale
$\xi$	Paramètre de pondération sur $V_Z$

## ANNEXE B

## Mouvement optimal

Il est donné, à chaque instant  $k$  ( $k$  est omis dans les équations par souci de simplicité) par :

$$\left\{ \begin{array}{l} V_X^* = \frac{1}{v} (-\lambda B + \lambda C V_Z^* + (1 - \lambda) V_X^N) \\ V_Y^* = \frac{1}{v} (-\lambda A + \lambda D V_Z^* + (1 - \lambda) V_Y^N) \\ V_Z^* = \frac{\lambda (-E + \frac{1}{v} |\lambda(BC+AD) - (1 - \lambda)(V_X^N + D V_Y^N)|)}{\lambda(1 - F + \frac{\lambda}{2}(C^2 + D^2)) - \xi - 1} - \frac{(1 - \lambda) V_Z^N}{\lambda(1 - F + \frac{\lambda}{2}(C^2 + D^2)) - \xi - 1} \end{array} \right.$$

$$A = \sum_{i=1}^{N(k)} \frac{\sigma_i \tilde{\lambda}_1^i}{\tilde{Z}_i} ; \quad B = \sum \frac{\sigma_i \tilde{\lambda}_2^i}{\tilde{Z}_i} v_i^d$$

$$C = \sum \sigma_i \frac{\tilde{x}_1^i}{\tilde{Z}_i^2} ; \quad D = \sum \sigma_i \frac{\tilde{y}_1^i}{\tilde{Z}_i^2}$$

$$E = \sum \sigma_i \left( \frac{\tilde{\lambda}_2^i \tilde{x}_1^i + \tilde{\lambda}_1^i \tilde{y}_1^i}{\tilde{Z}_i} \right) v_i^d$$

$$F = \sum \sigma_i \left( \frac{\tilde{x}_1^2 + \tilde{y}_1^2}{\tilde{Z}_i^2} \right)$$

$$\mu = \sum \sigma_i \frac{1}{\tilde{Z}_i^2} ; \quad v = \lambda \mu + 1 - \lambda$$

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique

